

Задача А. Две самые далёкие точки (наивная версия)

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 0.25 секунд
Ограничение по памяти: 256 мегабайт

Формат входных данных

Первая строка содержит количество точек N , ($1 \leq N \leq 7000$). Каждая из последующих N строк содержит два целых числа — координаты x_i и y_i . Координаты по модулю не превосходят 10^9 .

Формат выходных данных

Выведите в выходной файл расстояние между двумя наиболее удалёнными точками с максимально возможной точностью.

Примеры

стандартный ввод	стандартный вывод
7 0 0 1 1 2 2 0 2 1 3 0 1 2 0	3.16227766016837952279
8 0 0 38 85 40 90 20 96 0 100 -20 96 -40 90 -38 85	100.00000000000000000000

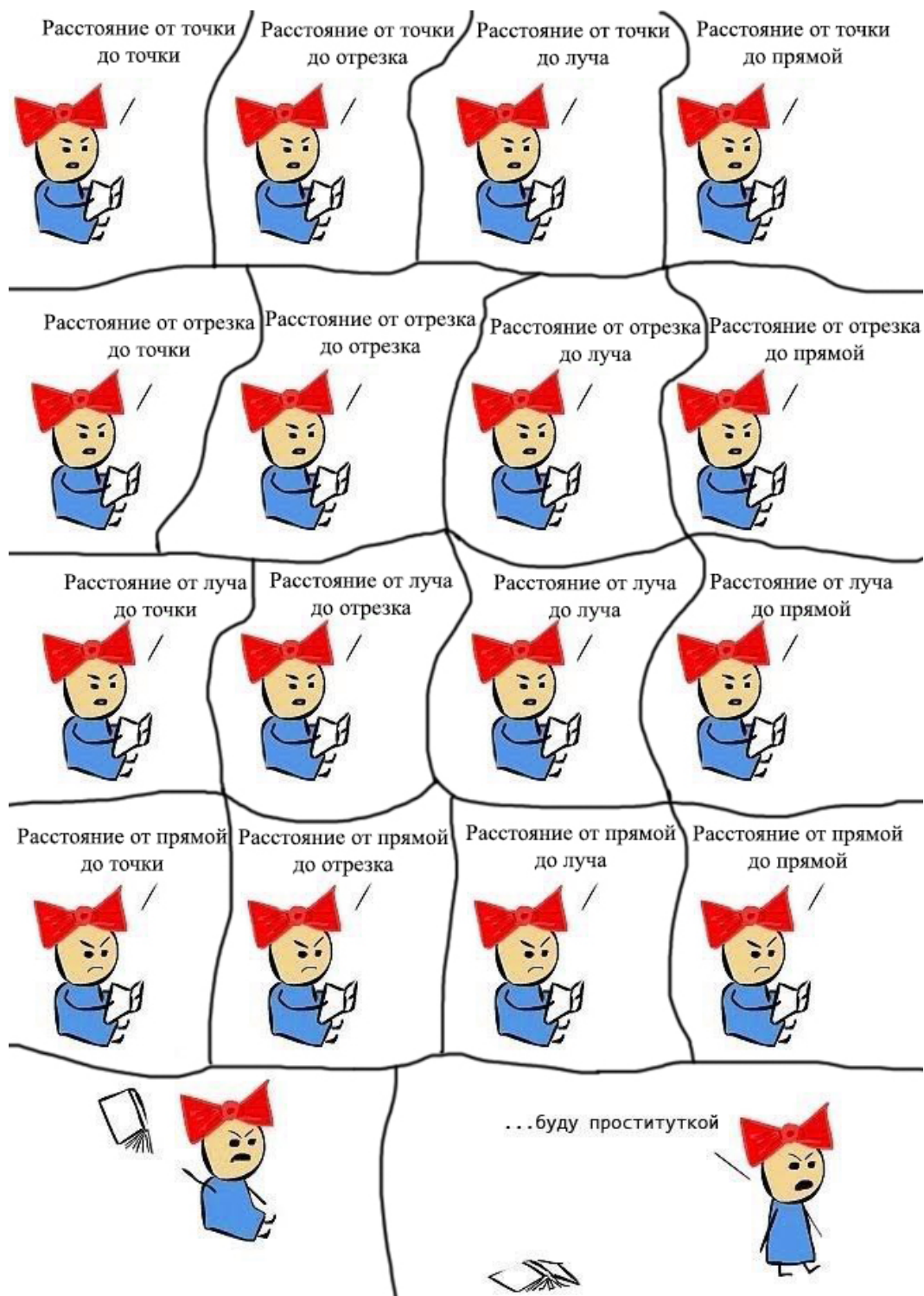
Замечание

Не используйте хитрые алгоритмы, напишите наивное решение за $O(n^2)$

Задача В. 16

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Даны 4 точки A, B, C, D.
Посчитайте:



Даны 4 точки A, B, C, D.

Посчитайте:

- Расстояние от точки A до точки C.
- Расстояние от точки A до отрезка CD.
- Расстояние от точки A до луча CD.
- Расстояние от точки A до прямой CD.
- Расстояние от отрезка AB до точки C.
- Расстояние от отрезка AB до отрезка CD.
- Расстояние от отрезка AB до луча CD.
- Расстояние от отрезка AB до прямой CD.
- Расстояние от луча AB до точки C.
- Расстояние от луча AB до отрезка CD.
- Расстояние от луча AB до луча CD.
- Расстояние от луча AB до прямой CD.
- Расстояние от прямой AB до точки C.
- Расстояние от прямой AB до отрезка CD.
- Расстояние от прямой AB до луча CD.
- Расстояние от прямой AB до прямой CD.

Формат входных данных

Даны координаты четырех точек, по одной точке в строке: $X_a, Y_a, X_b, Y_b, X_c, Y_c, X_d, Y_d$. Все числа целые, по модулю не превосходят 10000.

Формат выходных данных

Выведите 16 чисел по одному в строке. Числа в ответе должны быть выданы с точностью не менее 6 знаков после десятичной точки.

Пример

стандартный ввод	стандартный вывод
1 2	5.6568542495
7 1	5.6000000000
5 6	5.6000000000
8 2	5.6000000000
	4.6031716446
	1.4142135624
	1.4000000000
	1.4000000000
	4.6031716446
	1.1507929111
	0.0000000000
	0.0000000000
	4.6031716446
	1.1507929111
	0.0000000000
	0.0000000000

Задача С. Векторы

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Даны два ненулевых вектора. Требуется вычислить:

- Длину первого и второго вектора (два числа)
- Вектор, образованный сложением данных двух векторов
- Скалярное и векторное произведения данных векторов
- Площадь треугольника, построенного из этих векторов
- Полярные углы, которые образуют данные радиус-векторы
- Угол между заданными векторами
- Часть плоскости, которую занимает сектор, образованный двумя данными векторами.

Формат входных данных

В двух строках входного файла заданы по четыре числа — координаты начала и конца первого вектора, затем второго. Все числа целые, по модулю не превосходящие 10000.

Формат выходных данных

В каждой строке выходного файла — ответ на соответствующий пункт задачи с точностью не менее 10^{-6} .

Примеры

stdin	stdout
5 1 2 6	5.830952 9.219544
1 1 7 8	3 12
	17 -51
	25.500000
	2.111216 0.862170
	5.034140
	0.801208

Задача D. Базовые операции над вектором

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Даны две несовпадающих точки на плоскости, требуется вычислить:

- Вектор с началом в первой и концом во второй точках
- Соответствующий ему нормированный вектор
- Вектор, сонаправленный первому и имеющий заданную длину
- Вектор, полученный путем поворота первого вектора на 90° по часовой стрелке
- Вектор, полученный путем поворота первого вектора на 90° против часовой стрелки

Формат входных данных

В первых двух строках даны по два целых числа, — координаты заданных точек. В третьей строке записано натуральное число, — длина, которую должен иметь построенный в третьем пункте задачи вектор.

Все числа во входном файле по модулю не превосходят 1000.

Формат выходных данных

В отдельных строках выходного файла нужно вывести координаты векторов, соответствующих каждому из пунктов задачи, с точностью до 10^{-4} .

Примеры

stdin	stdout
3 6	0 2
3 8	0.000000 1.000000
5	0.000000 5.000000
	2 0
	-2 0

Задача Е. Точка в углу

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Дан угол AOB (O - вершина угла, A и B - точки на сторонах) и точка P . Определите, принадлежит ли точка P углу AOB (включая его стороны: лучи OA и OB).

Формат входных данных

Программа получает на вход координаты точек A , O , B , P . Все координаты - целые, не превосходят 10^5 по модулю. Точки A , O , B не лежат на одной прямой.

Формат выходных данных

Программа должна вывести слово *YES* или *NO*.

Примеры

стандартный ввод	стандартный вывод
0 1 0 0 1 0 1 1	YES
1 0 0 0 0 1 -1 -1	NO
-1 5 3 3 0 0 -1 1	YES

Задача F. Пересечение отрезков

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Даны два отрезка: AB и CD . Определите, какое множество точек является пересечением этих отрезков.

Формат входных данных

Программа получает на вход восемь целых чисел, по абсолютной величине не превосходящих 10^4 — координаты точек A, B, C, D . Точки могут совпадать (в том числе могут совпадать и концы одного отрезка).

Формат выходных данных

Если указанные отрезки не пересекаются, то выведите строку «Empty». Если отрезки пересекаются в одной точке, то выведите два числа — координаты точки пересечения. Если пересечением является отрезок, то выведите четыре числа — координаты двух концов отрезка в лексикографическом порядке (то есть сначала нужно вывести ту точку, у которой меньше координата x , а если у них равны координаты x , то ту, у которой меньше координата y). Все числа следует выводить с точностью не менее 6 знаков после запятой.

Примеры

<code>stdin</code>	<code>stdout</code>
0 0 9 9 9 5 0 5	5.0000000000 5.0000000000
0 0 9 9 15 15 7 7	7.0000000000 7.0000000000 9.0000000000 9.0000000000
0 0 9 9 10 10 10 10	Empty

Задача G. Пересечение прямых

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

На плоскости даны две прямые. Каждая прямая задается парой точек, через которые она проходит. Требуется установить, пересекаются ли эти прямые, и найти координаты точки пересечения.

Формат входных данных

Вводятся сначала координаты двух различных точек, через которые проходит первая прямая, а затем - координаты еще двух различных (но, быть может, совпадающих с первыми двумя) точек, через которые проходит вторая прямая. Координаты каждой точки - целые числа, по модулю не превышающие 10^3 .

Формат выходных данных

Если прямые не пересекаются, выведите одно число 0. Если прямые совпадают, выведите 2. Если прямые пересекаются ровно в одной точке, то выведите сначала число 1, а затем два вещественных числа - координаты точки пересечения с точностью до 10^{-6} .

Примеры

стандартный ввод	стандартный вывод
0 0 1 1 1 0 -1 2	1 0.50000 0.50000
1 17 5 20 4 40 2 18	1 1.97561 17.73171
2 2 3 3 5 5 10 10	2

Задача Н. Расстояние от точки до прямой

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Найдите расстояние от заданной точки до заданной прямой.

Формат входных данных

Пять целых чисел — координаты точки и коэффициенты A , B и C нормального уравнения прямой.

Формат выходных данных

Одно число — расстояние от точки до прямой с точностью не менее 10^{-6} .

Примеры

<code>stdin</code>	<code>stdout</code>
1 1 1 1 -1	0.7071067812

Задача I1. Принадлежность точки прямой

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Формат входных данных

Пять чисел — координаты точки и коэффициенты A , B и C нормального уравнения прямой.

Формат выходных данных

Одна строка «YES», если точка принадлежит прямой, и «NO» в противном случае.

Примеры

<code>stdin</code>	<code>stdout</code>
3 7 -2 1 -1	YES

Задача I2. Принадлежность точки лучу

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Формат входных данных

Шесть чисел — координаты точки и координаты начала и конца вектора.

Формат выходных данных

Одна строка «YES», если точка принадлежит лучу, определяемому вектором, и «NO» в противном случае.

Примеры

стандартный ввод	стандартный вывод
1 6 3 7 5 8	NO
4 2 4 2 4 5	YES

Задача I3. Принадлежность точки отрезку

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Формат входных данных

Шесть целых чисел от -10^4 до 10^4 — координаты точки и координаты концов отрезка.

Формат выходных данных

Одна строка «YES», если точка принадлежит отрезку, и «NO» в противном случае.

Примеры

стандартный ввод	стандартный вывод
3 3 1 2 5 4	YES
4 2 4 2 4 5	YES

Задача J1. Пусти козла в огород - 1

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Однажды на огород к Ивану Петровичу забежало целых три козла. Известно, что козлы весьма агрессивные животные, особенно когда речь идет о борьбе за вкусную капусту. Поэтому каждый из трех козлов, заметив других козлов, замер на месте и начал наблюдать за оставшимися козлами: одним глазом за одним козлом, другим — за другим. Естественно, для этого козлу нужно “косить” глазами.

Определите наибольший угол, на который пришлось “раскосить” глазами козлам.

Формат входных данных

Программа получает на вход координаты трех точек, в которых стоят козлы (сначала координаты первого козла, затем второго, затем третьего). Координаты — пара целых чисел, не превосходящих 10^4 по модулю.

Формат выходных данных

Выведите единственное число — величину самого большого угла, на который “косят” глаза козлов с точностью не меньше 6 знаков после запятой.

Пример

стандартный ввод	стандартный вывод
0 0 3 0 0 4	90.000000

Задача J3. Пусти козла в огород - 3

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

64 мегабайта

Козла пустили в огород. К несчастью для козла, его крепко-накрепко привязали к точке C крепчайшей альпинистской верёвкой длины L (то есть козёл может кушать травку, удалённую от точки C не более чем на L метров). И только одно скрасило жизнь бедного козла — недалеко от него виднеется грядка так любимых им ананасов, представляющая собой отрезок прямой с концами в точках A и B . Но, при большом старании, козёл может растянуть крепчайшую альпийскую верёвку. Определите, на какую длину козлу придётся растянуть верёвку, чтобы добраться хотя бы до одного ананасика? А чтобы съесть все ананасы?

Формат входных данных

На входе находятся координаты точек A , B , C и длина верёвки L . Все числа целые, $L \geq 0$, все координаты не превосходят по модулю 10000. Числа разделены пробелами или переводами строк.

Формат выходных данных

В первой строке вывести минимальную длину, на которую козлу придётся растянуть верёвку, чтобы дотянуться до грядки с ананасами. Во второй строке вывести минимальную длину, на которую козлу придётся растянуть верёвку, чтобы добраться до всех ананасов с грядки. Все числа выводить с точностью не менее 6 знаков после запятой.

Пример

стандартный ввод	стандартный вывод
8 -6 8 6	1.0000000000
0 0 7	3.0000000000

Задача К. Уравнение прямой I

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Формат входных данных

Четыре числа — целые координаты двух различных точек на прямой. ($|x|, |y| \leq 10^4$)

Формат выходных данных

Три числа — дробные коэффициенты A , B и C уравнения этой прямой с точностью до шестого знака после запятой. Выводите результат с точностью не менее 6 знаков после запятой.

Примеры

<code>stdin</code>	<code>stdout</code>
1 2 3 1	-1 -2 5

Задача L. Биссектриса

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Найдите прямую, содержащую биссектрису угла, заданного вершиной X и двумя точками Y и Z на его сторонах.

Формат входных данных

Шесть целых чисел, не превышающих по модулю 10^4 — координаты точек X , Y и Z .

Формат выходных данных

Три числа — коэффициенты нормального уравнения биссектрисы угла $\angle YXZ$ с точностью до шести знаков после запятой.

Пример

стандартный ввод	стандартный вывод
1 1 1 0 0 1	-1.0 1.0 -0.0

Задача М. Река

Имя входного файла:	<code>stdin</code>
Имя выходного файла:	<code>stdout</code>
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Каждый день в Монастыре Светлой Луны проходит утренний обряд, являющийся важнейшим элементом жизненного пути каждого монаха. Во время этого обряда один из монахов должен выйти из Монастыря, пройти к реке Мацанг (истоку Брахмапутры) и, набрав там воды, принести ее в Храм Солнца Дхармы, расположенный неподалеку от Монастыря. При этом обряд должен быть закончен ровно к рассвету. Для того чтобы достичь такой точности, монахам необходимо знать длину кратчайшего пути от Монастыря до Храма, проходящего через реку Мацанг. Вы призваны помочь Монастырю Светлой Луны рассчитать эту самую длину. Поскольку река близ Монастыря не имеет сколько-нибудь значительных изгибов, вы можете считать ее идеально прямой. Кроме того, можно пренебречь ее шириной. Однако необходимо помнить, что Мацанг — река горная и опасная, а поэтому перейти ее вброд абсолютно невозможно! При этом, естественно, ни Монастырь, ни Храм не находятся в реке.

Формат входных данных

В первой строке записаны координаты Монастыря, во второй — координаты Храма. В третьей строке записаны числа k, b — коэффициенты, задающие уравнение реки ($y = kx + b$). Все числа кроме k — целые и не превосходят по модулю 10000. Коэффициент k — действительное число, не превосходящее по модулю 100.

Формат выходных данных

Выведите одно действительное число, округленное до трех знаков после запятой — длину кратчайшего пути. В том случае, если Монах не может пройти к Храму, необходимо вывести "No solution."

Примеры

stdin	stdout
0 10 20 10 0 0	28.284
0 10 20 -10 0 0	No solution.

Задача N. Полярный угол

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Формат входных данных

Два числа — координаты точки ($|x|, |y| \leq 10^4$), не совпадающей с началом координат.

Формат выходных данных

Одно число — величина её полярного угла в радианах из интервала $[0, 2\pi)$. Все числа следует выводить с точностью не менее 6 знаков после запятой.

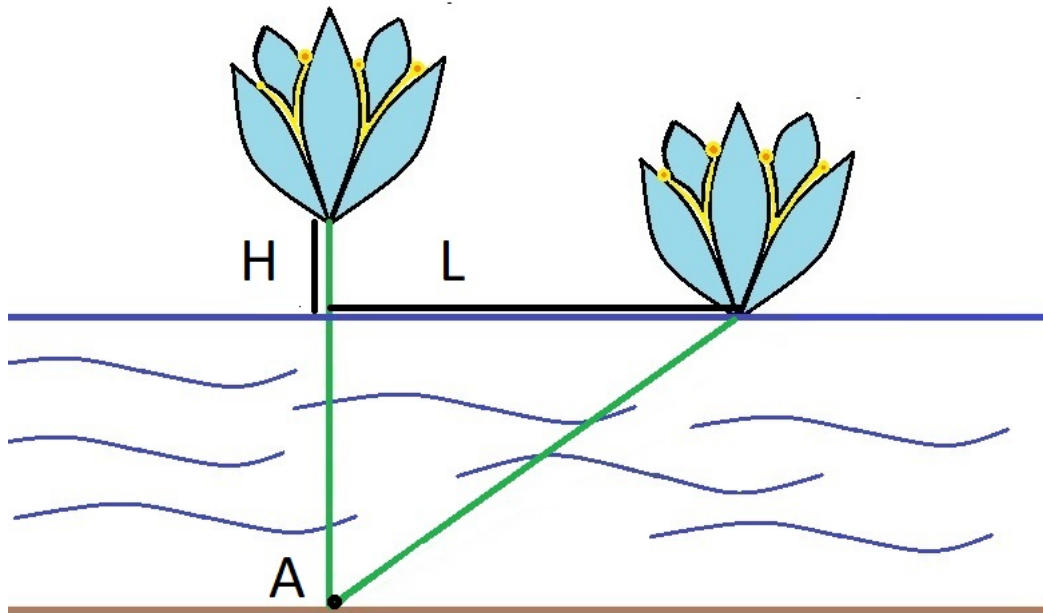
Примеры

<code>stdin</code>	<code>stdout</code>
2 3	0.98279372324732906796

Задача О. Водяная лилия

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Инесса плывала по озеру в лодке и заметила прекрасный цветок водяной лилии над поверхностью воды. Она подплыла к цветку поближе и оказалось, что цветок находится на высоте H над уровнем воды. Инесса взяла цветок и отплыла на расстояние L . В этот момент цветок коснулся воды.



Предположим, лилия растёт из точки A на дне озера, а стебель лилии всегда является отрезком прямой, один из концов которого закреплён в точке A . Предположим также, что изначально цветок находился прямо над точкой A , то есть стебель был вертикальным. Можете ли вы определить глубину дна озера в точке A ?

Формат входных данных

В единственной строке ввода записаны два целых числа H и L ($1 \leq H < L \leq 10^6$).

Формат выходных данных

Выведите одно действительное число — глубину озера в точке A . Абсолютная или относительная погрешность ответа не должна превышать 10^{-6} .

Более формально, пусть A — ваш ответ, B — ответ жюри. Тогда ваш ответ будет считаться правильным, если $\frac{|A-B|}{\max(1, |B|)} \leq 10^{-6}$

Примеры

стандартный ввод	стандартный вывод
1 2	1.50000000000000
3 5	2.66666666666667

Задача Р. Штрафы

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Новый градоначальник города Глупова решил с целью пополнения бюджета и экономии горючего провести кампанию борьбы с левым уклоном и левыми рейсами. Для этого он запретил водителям выполнять левые повороты, установив штраф за каждый поворот налево в размере одного миллиона (разворот поворотом налево не считается).

От тяжелого прошлого Глупову достались улицы, которые могут пересекаться под любыми углами. Градоначальник приказал установить компьютерную систему тотальной слежки, которая следит за каждым автомобилем, записывая его координаты каждый раз, когда тот меняет направление движения (включая начальную и конечную точки пути).

Требуется написать программу, вычисляющую по записанной последовательности координат автомобиля штраф, который должен быть взыскан с водителя.

Формат входных данных

В первой строке вводится целое число N - количество записанных пар координат ($1 \leq N \leq 1000$). В каждой из следующих N строк записана очередная из этих пар (вещественные числа).

Формат выходных данных

Выведите суммарный штраф водителя в миллионах.

Примеры

стандартный ввод	стандартный вывод
4 0 0 1 0 1 1 2 1	1

Задача Q. Ежик и грибы

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Ежик путешествует по координатной плоскости. Изначально он находится в точке с координатами $(0, 0)$ и смотрит в направлении оси абсцисс (оси Ox).

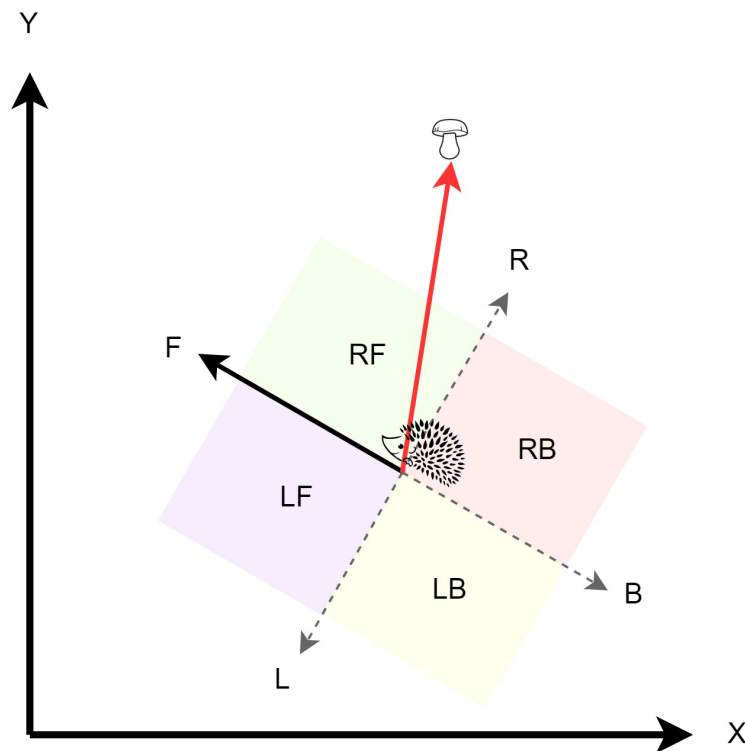
Цель ежика — собрать ровно n грибов, причем строго в описанном порядке, то есть сначала первый, затем второй, \dots , и только в конце n -й. Известно, что i -й гриб растет в точке с координатами (x_i, y_i) .

Чтобы собрать i -й гриб, ежик перемещается из своего текущего положения в точку (x_i, y_i) **строго по прямой**, все время смотря вперед по направлению движения. Определить точное направление движения ежик справится и сам с помощью нюха, но для этого ему сначала придется повернуться хотя бы приблизительно в нужную сторону. Помогите ему в этом!

Более формально, для каждого из n перемещений ежика выведите одну из строк:

- «L» или «R», если для перемещения к следующему грибу ежику следует повернуться ровно на 90° против или по часовой стрелке соответственно;
- «F» или «B», если для перемещения к следующему грибу не требуется поворачиваться вообще или требуется развернуться на 180° соответственно;
- «LF», «LB», «RB» или «RF», если следующий гриб находится строго внутри первой (слева-спереди), второй (слева-сзади), третьей (справа-сзади) или четвертой (справа-спереди) четверти плоскости относительно текущего направления ежика.

Ниже проиллюстрировано, как изменяется направление ежика. Предположим, он только что съел гриб и смотрит в направлении «F». Красным цветом указано направление до следующего гриба, поэтому ежу нужно далее следовать в направлении четвертой четверти «RF».



Формат входных данных

В первой строке дано единственное целое число n — количество грибов, которые ежик хочет собрать ($1 \leq n \leq 10^5$).

Во i -й из следующих n строк через пробел даны два целых числа x_i и y_i — координаты i -го гриба ($|x_i|, |y_i| \leq 10^9$). Гарантируется, что никакие два гриба не растут в одной точке, и что никакой гриб не растет в точке $(0, 0)$.

Формат выходных данных

Выведите информацию об n поворотах, совершаемых ежиком, в формате, описанном в условии. Информацию о последовательных поворотах разделяйте пробелом или переводом строки.

Примеры

стандартный ввод	стандартный вывод
3 1 1 0 2 -3 -1	LF L L
7 11 0 7 3 3 3 -6 -4 10 100 -1 -10 2 20	F LB LF LF RB RB B